Numpy Cheat Sheet

```
import numpy as np

```

Creating Arrays

```
a = np.array([1,2,3])
b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
c = np.array([[(1.5,2,3), (4,5,6)],[(3,2,1), (4,5,6)]], dtype = float)

```

Initial Place Holder

```
np.zeros((3,4))
np.ones((2,3,4),dtype=np.int16)
d = np.arange(10,25,5)
np.linspace(0,2,9)
e = np.full((2,2),7)
f = np.eye(2)
np.random.random((2,2))
np.empty((3,2))

```

```
array([[1.39069238e-309, 1.39069238e-309],
       [1.39069238e-309, 1.39069238e-309],
       [1.39069238e-309, 1.39069238e-309]])
```

I/O

Saving and Loading on disk

```
np.save('my_array' , a)
np.savez( 'array.npz', a, b)
np.load( 'my_array.npy')
```

```
array([1, 2, 3])
```

Saving and Loading Text Files

```
np.loadtxt("myfile.txt")
np.genfromtxt("my_file.csv", delimiter= ',')
np.savetxt( "myarray.txt", a, delimiter= " ")

```

Asking for Help

```
np.info(np.ndarray.dtype)

```

```
    Data-type of the array's elements.

    Parameters
    ----------
    None

    Returns
    -------
    d : numpy dtype object

    See Also
    --------
    numpy.dtype

    Examples
    --------
    >>> x
    array([[0, 1],
           [2, 3]])
    >>> x.dtype
    dtype('int32')
    >>> type(x.dtype)
    <type 'numpy.dtype'>
```

Inspecting your Array

```
a.shape #Array dimensions
len(a)#Length of array
b.ndim #Number of array dimensions
e.size #Number of array elements
b.dtype  #Data type of array elements
b.dtype.name  #Name of data type


```

```
    'float64'
```

Data Types

```
np.int64 #Signed 64-bit integer types
np.float32
np.complex #Complex numbers represented by 128 floats
np.bool  #Boolean type storing TRUE and FALSE values
np.object #Python object type
np.string_ #Fixed-length string type
np.unicode_ #Fixed-length unicode type

```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DeprecationWarning: `
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  This is separate from the ipykernel package so we can avoid doing imports until
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: DeprecationWarning: `
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: `
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  """
numpy.str_
```

Array Mathematics

Arithmatic Operations

```
g = a - b #Subtraction
np.subtract(a,b) #Subtraction
b + a #Addition
np.add(b,a) #Addition
a/b #Division
np.divide(a,b) #Division
a * b #Multiplication
np.multiply(a,b) #Multiplication
np.exp(b) #Exponentiation
np.sqrt(b) #Square root
np.sin(a)  #Print sines of an array
np.cos(b) #Elementwise cosine
np.log(a)#Elementwise natural logarithm
e.dot(f) #Dot product
```

```
array([[7., 7.],
       [7., 7.]])
```

Comparison

```
a == b #Elementwise comparison
a< 2 #Elementwise comparison
np.array_equal(a, b) #Arraywise comparison
```

```
False
```

Copying Arrays

```
h = a.view()#Create a view of the array with the same data
np.copy(a) #Create a copy of the array
h = a.copy() #Create a deep copy of the array

```

Sorting Arrays

```
a.sort() #Sort an array
c.sort(axis=0) #Sort the elements of an array's axis
```

Subsetting, Slicing, Indexing

Subsetting

```
a[2] #Select the element at the 2nd index
b[1,2] #Select the element at row 1 column 2(equivalent to b[1][2])
```

```
6.0
```

Slicing

```
a[0:2]#Select items at index 0 and 1
b[0:2,1] #Select items at rows 0 and 1 in column 1
b[:1]
#Select all items at row0(equivalent to b[0:1, :])
c[1,...] #Same as[1,:,:]
a[ : : -1] #Reversed array a array([3, 2, 1]
```

```
array([3, 2, 1])
```

Boolean Indexing

```
a[a<2] #Select elements from a less than 2

```

```
array([1])
```

Array Manipulation

Transposing Array

```
i = np.transpose(b) #Permute array dimensions
i.T #Permute array dimensions
```

```
array([[1.5, 2. , 3. ],
       [4. , 5. , 6. ]])
```

Changing Array Shape

```
b.ravel() #Flatten the array
g.reshape(3, -2) #Reshape, but don’t change data

```

```
array([[-0.5,  0. ],
       [ 0. , -3. ],
       [-3. , -3. ]])
```

Adding/Removing Elements

```
np.append(h,g) #Append items to an array
np.insert(a,1,5)  #Insert items in an array
np.delete(a,[1])  #Delete items from an array

```

```
array([1, 3])
```

Splitting Arrays

Splitting Arrays

```
np.hsplit(a,3) #Split the array horizontally at the 3rd index
np.vsplit(c,2) #Split the array vertically at the 2nd index
```

```
[array([[[1.5, 2. , 1. ],
         [4. , 5. , 6. ]]]), array([[[3., 2., 3.],
         [4., 5., 6.]]])]
```